Manual técnico Analizador de Torneos

Clases:

Scanner:

Esta clase implementa un lexer que convierte un texto de entrada en tokens para su análisis sintáctico

En su constructor normalizamos los saltos de lines, creamos un diccionario de palabras reservadas, llevamos la cuenta de la line y columnas de los tokens

```
constructor(input) {
    this.input = input.replace(/\r\n/g, '\n') + '\0';
    this.pos_char = 0;
    this.buffer = '';
    this.char_line = 1;
    this.char_col = 1;
    this.next_char = '';
    this.keywords = {
        TORNEO: 'KW_torneo',
        EQUIPOS: 'KW_equipos',
        equipo: 'KW_equipo',
        jugador: 'KW_jugador',
        ELIMINACION: 'KW_eliminacion',
        fase: 'KW_fase',
        partido: 'KW_partido',
        goleador: 'KW_goleador',
        Posiciones: 'KW_posiciones'
    }
}
```

Funciones:

initBuffer:

inicializamos un buffer con un carácter y actualiza contadores de posiciones

```
initBuffer(current_char) {
    this.buffer = current_char;
    this.char_col ++;
    this.pos_char ++;
    this.last_char = current_char;
}
```

addBuffer:

añade un carácter al buffer y actualiza contadores

```
addBuffer(current_char) {
    this.buffer += current_char;
    this.char_col ++;
    this.pos_char ++;
    this.last_char = current_char;
}
```

S0 (Estado inicial):

función principal que actúa como punto de entrada para el análisis léxico, implementa un autómata finito determinista

- Identifica el tipo de carácter y dirige al estado correspondiente
- Ignora espacios, tabas y saltos de línea
- Detectar errores léxicos con mensajes informativos
- Retorna un EOF (End Of File – fin del archivo) cunado llega al fin del archivo

```
S0() {
    while((this.next_char = this.input[this.pos_char]) !== '\0') {
        this.next_char = this.input[this.pos_char]
        if( // TRANSICIONES PARA
            // A-Z
            this.next_char.charCodeAt(0) >= 65 && this.next_char.charCodeAt(0) <= 90 ||
            // a-z
            this.next_char.charCodeAt(0) >= 97 && this.next_char.charCodeAt(0) <= 122
        ) {
            this.initBuffer(this.next_char);
            return this.S1();
        }

        if(this.next_char === '"') { // TRANSICIONES PARA CADENAS
            this.initBuffer(this.next_char);
            return this.S2();
        }

        if(this.next_char === '{') {
            this.initBuffer(this.next_char);
            return this.S4();
        }
```

Estados del autómata S1-S9

Cada estado representa una categoría léxica

```
S1() {
    this.next_char = this.input[this.pos_char]
    if(
        // A-Z
        this.next_char.charCodeAt(0) ≥ 65 && this.next_char.charCodeAt(0) ≤ 90 ||
        // a-z
        this.next_char.charCodeAt(0) ≥ 97 && this.next_char.charCodeAt(0) ≤ 122
    ){
        this.addBuffer(this.next_char);
        return this.S1();
    }

    // Retorna el token reconocido
    return { lexeme: this.buffer, type: this.keywords[this.buffer] || 'TK_id', line: this.char_line, column: this.char_col };
}

S2() {
    this.next_char = this.input[this.pos_char]
    if(this.next_char ≠ '"') {
        this.addBuffer(this.next_char);
        return this.S2();
    }

    this.addBuffer(this.next_char);
    return this.S3();
}
```

- S1: identificadores y palabras reservadas
- S2-S3: cadenas entre comillas
- S4: llave izquierda {
- S5: Llave derecha }
- S6: Corchete izquierdo [
- S7: Corchete derecho]
- S8: Coma ,
- S9: Dos puntos :

Next_token:

Función publica que devuelve el siguiente token del input

```
next_token = () ⇒ this.S0();
```

Indexs:

Index.html:

Este archivo HTML implementa una interfaz web para el analizar de torneos

Head:

Define la codificación y vista para dispositivos móviles, importa la hoja de estilos y archivos CSS locales

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>TourneyJS - Analizador de Torneos</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/codemirror.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/theme/monokai.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/theme/material.min.css">
    <link rel="stylesheet" href="./Interfaz/styles.css">
</head>
```

Body:

Organiza toda la interfaz, el área de contenido dividida en 2 paneles, incluye los archivos JS del proyecto, muestra los botones de navegación Carga de archivo, Analizar Torneo, Generar Reportes, Mostrar Bracket

```
<main class="main-content">
    <div class="content-section">
        <div class="editor-container">
            <div class="editor-panel">
                <div class="editor-header">
                    Editor de Código - Configuración del Torneo      You,
                </div>
                <div class="editor-wrapper">
                    <textarea id="codeEditor"></textarea>
                </div>
            </div>

            <div class="results-panel">
                <div class="results-header">
                    Resultados del Análisis
                </div>
                <div class="results-content" id="resultsContent">
                    <div class="empty-state">
                        <div class="empty-icon"></div>
                        <p></p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</main>
```

```
<!-- CodeMirror JavaScript -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/codemirror.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/mode/javascript/javascript.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/addon/edit/closebrackets.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/addon/edit/matchbrackets.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/codemirror/5.65.2/addon/lint/lint.min.js"></script>
<script src="./Lenguaje/Scanner.js"></script>
<script src="./Lenguaje/index.js"></script>
```

```
<nav class="nav-buttons">
    <button class="nav-btn" onclick="document.getElementById('fileInput').click()">Cargar Archivo</button>
    <button class="nav-btn" onclick="analizarTorneo()">Analizar Torneo</button>
    <button class="nav-btn" onclick="generarReporte()">Generar Reporte</button>
    <button class="nav-btn" onclick="mostrarBracket()">Mostrar Bracket</button>
</nav>
```

Index.js:

En este implementamos la funcionalidad principal de la interfaz web integrando el analizador léxico (Scanner)

Editor:

Almacena la instancia de CodeMirror para manipular el editor de código

DOMContentLoaded:

- Se ejecuta cuando el DOM (modelo de objetos del documento) esta completamente cargado
- Configura el editor de CodeMirror con las opciones mode, theme, lineNumbers, autoCloseBrackets, matchBrackets

```
document.addEventListener("DOMContentLoaded", function () {
    editor = CodeMirror.fromTextArea(document.getElementById("codeEditor"), {
        mode: "application/json",
        theme: "material",
        styleActiveLine: true,
        lineNumbers: true,
        lineWrapping: false,
        autoCloseBrackets: true,
        matchBrackets: true,
        indentUnit: 4,
        tabSize: 4,
        scrollbarStyle: "native",
    });

    setTimeout(() => {
        editor.refresh();
    }, 100);
});
```

cargarArchivos:

maneja la selección de archivos mediante el input oculto

```
function cargarArchivo(event) {
    const files = event.target.files;
    if (files.length > 0) {
        const file = files[0];

        const reader = new FileReader();

        reader.onload = function(e) {
            editor.setOption('value', e.target.result);
        };

        reader.onerror = function() {
            alert("Error al leer el archivo");
        };

        reader.readAsText(file);
    }
}
```

analizadorTorneo:

función principal que ejecuta el análisis léxico, obtiene el contenido del editor, crea una instancia del scanner y genera la tabla de todos los tokens encontrados

```
function analizarTorneo() {
    const contenido = editor.getValue();
    const scanner = new Scanner(contenido);
    const resultContent= document.getElementById('resultsContent');

   resultContent.innerHTML = `<div class="loading">Analizando</div>`;

    let tokenCount = 1;
    let token = scanner.next_token();

    const tabla = document.createElement('table');
    tabla.className = 'tabla-token';
    tabla.innerHTML = `
      <thead>
        <tr>
          <th>#</th>
          <th>Token</th>
          <th>Tipo</th>
          <th>Linea</th>
          <th>Columna</th>
          <th>Lexema</th>
        </tr>
      </thead>
      <tbody id="tokensBody"></tbody>
    `;
```

El analizador categoriza los tokens en:

Styles.css:

Hoja de estilos CSS que proporciona un diseño para la aplicación, modificando sus layout y contenedores, también dando cambios de diseño responsive como ajuste de tamaño de fuente y espaciado para móviles

```css
.loading {
  animation: pulse 1.5s ease-in-out infinite;
}

.tabla-token-container {
  max-height: 400px;
  overflow-y: auto;
  margin: 20px 0;
  border-radius: 8px;
  border: 1px solid #e1e5e9;
}

.tabla-token-container::-webkit-scrollbar {
  width: 8px;
}

.tabla-token-container::-webkit-scrollbar-track {
  background: #f1f3f4;
  border-radius: 4px;
}

.tabla-token-container::-webkit-scrollbar-thumb {
  background: #c1c8d0;
  border-radius: 4px;
}

.tabla-token-container::-webkit-scrollbar-thumb:hover {
  background: #a0a7b0;
}       You, 1 hour ago • se agregaro mas KW en la clase scanner
```

```css
@media (max-width: 768px) {
  .header h1 {
    font-size: 2em;
  }

  .nav-buttons {
    gap: 10px;
  }

  .nav-btn {
    padding: 10px 16px;
    font-size: 12px;
  }

  .section-header {
    padding: 15px 20px;
  }

  .section-title {
    font-size: 1.5em;
  }
}
```

```css
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen,
    Ubuntu, Cantarell, sans-serif;      You, yesterday • interfaz, entrada y scanner
  background: #8c8c8c;
  min-height: 100vh;
  padding: 20px;
}

.container {
  max-width: 1400px;
  margin: 0 auto;
}
```